Case Base Maintenance Approach *

Mohamed Karim HAOUCHINE, Brigitte CHEBEL-MORELLO, Noureddine ZERHOUNI

Laboratoire d'Automatique de Besançon, LAB –UMR CNRS 6596, France

24, rue Alain Savary, 25000 Besançon - France

Abstract

Case base Maintenance is an active Case Based Reasoning research area. The main stream focuses on the method for reducing the size of the case-base while maintaining case-base competence. This paper gives an overview of these works, and proposes a case deletion strategy based on competence criteria using a novel approach. The proposed method, even if inspired from existing literature, combines an algorithm with a Competence Metric (CM). A series of tests are conducted using two standards data-sets as well as a locally constructed one, on which, three Case Base Maintenance approaches were tested. This experimental study shows how this technique compares favourably to more traditional strategies across two standard data-sets.

Key words: Knowledge-based systems, case-based reasoning, case-base maintenance, competence, case deletion strategies

1 Introduction

Case Based Reasoning (CBR) is an approach to problem solving and learning, by reusing the solutions to similar problems stored as cases in a Case Base CB [1]. An experience is explained in completely structured formats or sizes such as objects or computer data in data-bases [13]. A case is an experience and an explanation during the episode of problem decision. Representation of cases is very diverse according to the nature of the task: diagnosis, planning, decision-making help, conception and so on [7]. However, case is composed of two options: a description of the situation that represents a "problem" and a solution used to solve the given problem in this situation [20]. A case is placed in a CB and is called source case in which one will get inspired to solve a new case that we will call target case. A CB is a collection of decision-making cases of the same problem [20]. A general CBR cycle may be described by five processes [21]:

1. DEVELOP the representation cases
2. RETRIEVE the most similar case or cases
3. REUSE the information and knowledge in that case to solve the problem
4. REVISE the proposed solution
5. RETAIN the parts of this experience likely to be useful for future problem solving

A new problem is solved by first, developing a target case that consists in developing case, by possibly supplementing the description of the new problem, then by retrieving one or more cases of the CB, reusing the case and revising the solution if need be, and retaining the new experience by adding it in the CB. Our field of study is diagnosis. For a case-based diagnosis system, the quality of case data is usually a key factor of the success of the system. In our research, we are facing a diagnosis problem in a product domain. We developed a case based reasoning system [9], and now our problem focused in the maintenance of this system. In effect, all mature information systems, more particularly; CBR systems need to be maintained while functioning, to guarantee the quality of the system. Indeed, the maintenance of the CBR system becomes necessary for all systems that are made to work for long periods and/or are developed to deal with large amounts of information

Email addresses: karim.haouchine@ens2m.fr (Mohamed Karim Haouchine), bmorello@ens2m.fr (Brigitte Chebel-Morello), zerhouni@ens2m.fr (Noureddine Zerhouni).

and cases. In the section 2, CBR systems, the objectives for having good quality systems and the different problems that can be seen over time and therefore the need to carry out maintenance operations will be detailed. Subsequently, a CBR study for maintenance will be explained in section 3. The CBR system arranges different knowledge sources, which can be maintained, the knowledge sources of the CB is of interest and therefore it will be elaborated in the fourth section. Finally, we will present the proposed method that consists of deleting cases after the competence criteria, before optimising the CB. The obtained results and also the comparison with different -already existing- methods will be presented in sections 7 and 8.

2 Historic of CBR systems

The CBR approach has been used in many industrial systems to solve problems in various implementation fields or domains. The development of CBR systems always bumps into an engineering knowledge problem. The representation and the organisation of cases, as well as specific knowledge that can take place during operations, cover an essential importance in the conception of a CBR system. To elaborate a relevant case that can reflect a common problem and expressed in coherent terms regarding the knowledge base and the adaptation of knowledge are difficult to form, the functionalities of training are limited and therefore these kinds of systems are often confronted to this difficulty [5]. A CBR system is a combination of processes and knowledge, often called "Knowledge containers" that allows to preserve and to exploit previous experiences [12]. Knowing that a knowledge base system, interested in the conception of CBR applications, also needs to be interested in the tools of knowledge representation. A Knowledge Base System (KBS) is generally seen as the automatic solution for human knowledge in a limited field of expertise [17]. It is acquired to have a Knowledge Base (KB) that represents the total of knowledge with the help of a representation-language of knowledge. KB takes part of an expert system that consists out of all information, in particular rules and facts, which constitute the competence field of the system [4]. The conception and the development of CBR systems are like every knowledge base systems, quite complex processes. Special environments are made to propose different methods and tools to guide conception, modelling and application of a CBR. One of the first CBR systems is CYRUS, which has been developed by Janet Kolodner. It is a system that retraces actions that revolve CYRUS Vance in its position as a state secretary [10]. Kolodner implements the ideas of SCHANK, which means a structure of dynamic memories for searching information. A lot of systems are therefore developed according to the SCHANK model, but for different tasks like CHEF, or JUDGE [11] or according to other models, for example a prototype model such as PROTOS [2]. CBR are also inspired by analogical reasoning and is considered as a particular case. The successive works or actions with CBR have worked in order to realise information tools to solve problems, to press one's finger entirely on this specific reasoning. If the first researches were the solution of cognitive psychology, present-day researches are plural-disciplinary. Actually, numerous works or actions tent to use CBRs to complete other methods such as systems based on rules or systems based on models [8], or even using neural networks [18]. These systems are known as hybrid systems. CBR supply a preliminary sketch of exploitable solutions by a different reasoning model that avoids constructing completely a solution in a potential immense space.

The majority of CBR systems explain problems concerning case-research phase as well as their adaptation that can be expensive in time. To come straight with their problems, maintenance of CBR systems become necessarily for all systems that are build to work for long periods of time and/or are developed to deal with large amounts of information and cases. The growing request of CBR system maintenance led to intensive researches in the concerned fields [24].

3 Maintenance of a Case-Based Reasoning System

The performance objectives for a CBR system provide criteria for evaluating the internal behaviour and task performance of a particular system for a given initial CB and sequence of a solved problem. The choice of case-base maintenance strategies is driven by the maintainer's performance goals for the system and by constraints on the system's design and the task environment [16]. In general, there will be multiple performance measures for a CBR system, and there is no guarantee that all of them can be maximized simultaneously. Smyth and McKenna define three types of top-level goals for CBR systems [32]:

1. Problem-solving efficiency goals (e.g., average problem-solving time)

2. Competence goals (the range of target problems solved)

3. Solution quality goals (e.g., the error level in solutions)

Therefore, a CBR system needs some sort of policies to achieve its maintenance goals. Its effectiveness depends on the speed and quality of the case base retrieval process, and it is related to the definition, the case representation, the CB organization, the various indexation used, and the definition of "good" similarities measurements for case search and the relation of retrieval-adaptation case. CB represents the memory and the main part of CBR systems, it can be flat or hierarchical, it contains elaborate cases which represent the experiments; an experiment constitutes a lesson which makes it possible of CBR system to achieve its goals. The information contained in a case depends on the applicability of objectives for which this case is built. In CB, we find cases which are composed of two parts, problem and solution parts. The CB is considered to be the core of any CBR system. This fact, pushed many research orientations, towards the involvement of CB Maintenance (CBM) [16]. The knowledge of CBR systems relates directly to cases, affected by changes in knowledge sources. Moreover, the CB source may be considered as that most sensitive knowledge source for CBR systems. Thus, consulting it may be the most appropriate approach to overtake maintenance operations. When the CB size increases, the search time, increases accordingly. This leads to the increase of problems resolution time, inducing to performance degradation [14].

4 Case-Base Maintenance

Case-base maintenance implements policies for revising the organization or contents (representation, domain content, accounting information, or implementation) of the CB in order to facilitate future reasoning for a particular set of performance objectives [14]. Note that this definition considers the information defining an indexing scheme to be an intrinsic organizational component of the CB itself. CBM may involve revising indexing information, links between cases, and/or other organizational structures and their implementations [29]. Maintenance in CBR may involve a number of different operations: out dated, redundant or inconsistent cases may be deleted; groups of cases may be merged to eliminate redundancy and improve reasoning power; cases may be re-described to repair incoherencies etc [29].

4.1 CBM Policies

CBM approach may be divided into two policies, one concerning optimization and the other partitioning of CB (See Figure 1). The CB approach aims to reduce case research time. This may be done, on one hand, following an optimization policy that deletes least relevant cases, using case addition or deletion strategies. On the other hand, following a partitioning policy, permitting the retrieval of distributed CBs using an attribute selection case scheme. Therefore, the attribute with rich-information content are selected, and may possess more potential to cover a wider CB structure [34]. The partitioning policy, allows the addition and deletion of cases in each small CB, without affecting the whole; one may cite several works using of dynamic or static neural network [18] [25]. However, there are many different ways to evaluate the quality of a CB. This is described in the following section.

4.2 Criteria for Evaluating Case Base

An "effective" CB is able to answer as many queries as possible efficiently and correctly. The criteria by which one can judge the effectiveness of a CB are given in [26], [27], [35]. Those of interest to this work are:

- *Competence*, measured by the range of problems that can be satisfactorily solved.
- *Performance*, measured by the answer time that is necessary to compute a solution for case targets. This measure is bound directly to adaptation and result costs.
- *Coverage* of a case is the set of target problems that it can be used to solve.

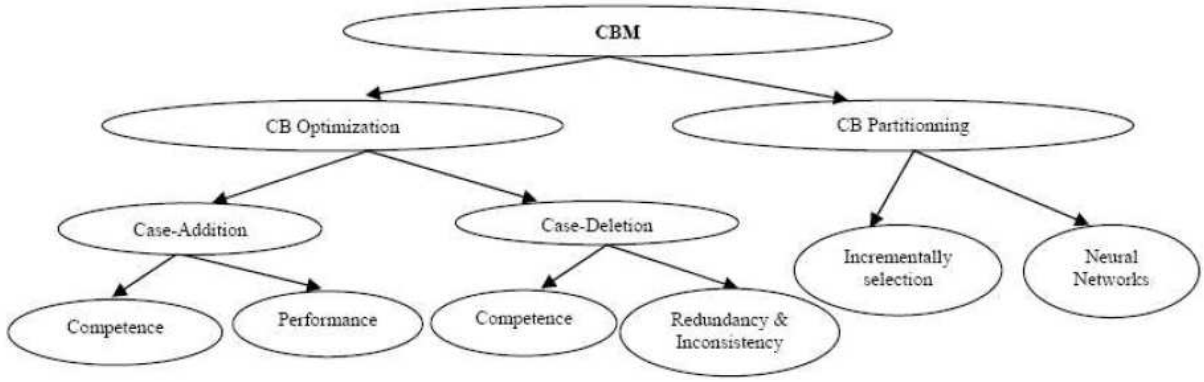


Fig. 1. Different strategies and criteria used in CBM

- *Reachability* of a target problem is the set of cases that can be used to provide a solution for the target.

After having given the definitions of the different criteria that permit the evaluation of a CB, we are going to see how they are used and estimated in the different CBM strategies. In the following paragraphs, two strategies will be developed; the addition-case and the deletion-case.

4.3 Case-addition Strategy

By the successive addition of cases to a virgin CB, reduced CB will be constructed, thus maximizing criteria. There are two methods, one maximizing the competence criterion, and the other the performance criterion.

4.3.1 Method maximizing the criteria of competence

Smyth and McKenna present a method that uses an explicit case competence model based on notions of coverage and reachability. Their "relative coverage" (RC) metric, provides a precise measurement of competence contributions for individual cases. The RC metric, associated with the condensed nearest-neighbour (CNN) algorithm, permits to successively retain only those cases which are not solved by a case that has already been retained, in order to obtain a new reduced CB [31]. This permits the selecting of cases, which have a big contribution concerning the CB recovery.

Q. Yang and J. Zhu describe a case-addition algorithm for CB compaction that uses a problem-neighbourhood model of case coverage. Cases are successively added based on added benefit/usefulness to the case set retained so far [35].

4.3.2 Method maximizing the criteria of performance

By analogy to the RC metric, Leake and Wilson developed a relative performance (RP) metric aimed at assessing the contribution of a case to the adaptation performance of the system [15]. To attain the benefit of adding the case to the CB, they first assume that the similarity metric will accurately select the most adaptable case for any problem. For each case that might be added to the CB, its contribution was estimated as regards to adaptation performance. The RP value for a case reflects how its contribution to adaptation performance compares to other cases. This metric can be used to guide case addition, favouring cases with low RP values. In the same manner, another metric was developed concerning a "performance benefit" (PB) metric estimating the actual numerical savings that the addition of each case provides. However, on one hand, the RC-CNN method provided a reduction rate of the CB size, better than the PR-CNN and PB-CNN methods. On the other hand, these two previous methods give a result concerning the adaptation cost of the CB cases better than RC-CNN.

4.4 Case-deletion Strategy

From a given CB, this strategy values case according to criteria in order to be able to suppress and bring the CB to a specified number of cases. The evaluation criteria such as competence, redundancy and inconsistency, have been used in different methods, which will be explained down bellow.

4.4.1 *Suppression Method for using case-base screening*

In this method, when the CB reaches a certain threshold it is screened entirely, usually followed by the process of case-deletion.

- *Random Deletion* is a very simple, inexpensive method and it is completely domain independent. Simply randomly select and delete a case from the CB once the CB size exceeds some predefined limit [19].

- *Ironically*, is a slightly more complicated method, it calculates the frequency of each case that is retrieved and it deletes cases which are not frequently accessed in the CB [22].

- *Utility Deletion (UD)* is based on Minton's utility metric which chooses a case item for deletion by estimating its performance benefits. This utility deletion method removes case items with negative utility [28]. The utility problem manifests itself as a trade-off between the solution quality associated with large CBs and the efficiency problem of working with a large CB. System efficiency is measured by taking the mean time to solve a target problem; note that the decreasing solution times correspond to an increase in efficiency. The solution quality is bound to the percentage of good answers, provided by the system. Solution quality increases with CB size [30].

- *Deletion based on redundancy and inconsistency* contains two modules of detection, one with redundancy and the other with inconsistency. After a series of test concerning each CB case, by these two modules, the specific cases can be removed -or kept- after the user approval [23].

- *Deletion based on CB size and density*, this deletion method was proposed by B. Smyth and M.T. Keane and it studies the size of the CB, the density and the distribution of cases in a CB. It tries to homogenize the case density [30].

4.4.2 *Method from categorization of the cases*

These methods, call on a modelling of the CB competence, proposed by B. Smyth and M.T. Keane [30], [32], [33] and [35]. Cases in the CB are categorized according to their competence. The key concepts in categorising cases are coverage and reachability.

Given a case-base C = {c1,…, cn} and c© is the set of target cases in the CB. Formally:

- Coverage(c) = {c©∈ C: Adaptable (c, c©)}, For c∈ C,
- Reachable(c) = {c©∈ C: Adaptable (c©, c)}, For c∈ C,

As a result, 4 categories of cases are considered:

- Pivotal Cases: Pivot(c) iff Reachable(c)-{c}= Ø
- Support Cases: Support(c) iff c©∈ Reachable(c)-{c}: Coverage(c©) ⊂ Coverage(c)
- Spanning Cases: Spanning(c) iff Pivot(c) Λ Coverage(c) ∩ $U_{c© \in \text{Reachable(c)-\{c\}}}$ Coverage(c) ≠ Ø
- Auxiliary Cases: Auxiliary (c) iff c©∈ Reachable(c)-{c}: Coverage(c) ⊄ Coverage(c©)

- *Footprint deletion*: this strategy works to remove irrelevant cases by guiding the CB towards an optimal configuration. Optimal in the sense that it maximises competence while minimising size. The case categories described above provide a means of ordering cases for deletion in terms of their competence contributions. Auxiliary cases are selected for deletion before support cases, which in turn are chosen before spanning and pivotal cases. The optimal CB can be constructed from all the pivotal cases plus one case from each support group. This strategy is not designed to eliminate the need for performance-based methods such as utility deletion [27].
- *Footprint Utility deletion*: is the hybrid strategy between footprint deletion and utility deletion. First, the footprint method is used to select candidates for deletion. If there is only one such candidate then it is deleted. If there are a number of candidates, therefore rather than selecting the one with the least coverage or the largest reachability set, the candidate with the lowest utility is chosen [27].

5 Methodology

Keeping in mind that no metric was developed in most case suppression approaches, a suppression method using a novel algorithm associated to a Competence Metric (CM) is proposed for CB optimisation. This method is a combination of Smyth and Keane approach [27] and the Relative Coverage metric [31].

$$\mathrm{RC}(c) = \sum_{c' \in CoverageSet(c)} \frac{1}{ReachabilitySet(c')} \tag{1}$$

Therefore, the developed metric is based on the notions of recovering and reachability. CM is defined as the individual contribution to a case as regards to its recovering group size by attributing two values, noted recovering value (Vr) and, reachability value (Va). *CompetenceMetric(c) = Vr(c)/Va(c)*

This metric is used as a benchmark for the suppression of cases within the CB. The cases which have a high CM value are favoured while the rest are dropped. The algorithm developed is as follows:

- Calculate the Vr and Va value of every case in the CB using the K Nearest Neighbour (KNN) algorithm with Euclidean norm, $dsc = [\sum_{i=1}^{p} (xs - xc)^2]^{1/2}$ (p represents the total number of attributes).
- Associate to each case its recovery unit and its reachability factor
- Determine the support cases and the support groups of each class and sort them according to their CM values
- Suppress the support cases with the lower CM values, keeping one case from each group with the most important CM
- Determine auxiliary inter and intra classes
- Suppress auxiliary cases according to their CM values
- Stop the algorithm if and only if each case covers its own case among existing ones, in its own class

---

Algorithm 1. Algorithm of the implemented method

This method aims at reducing the CB size by preserving maximum competence.

6 Experiments

The used method is based on a specific model of competence for case-based reasoning. We argue that it has more potential for guiding the construction of smaller CBs than some existing editing methods without compromising competence, specifically CNN on its own, CNN with NUN and RC distance ordering. In this section we compare the size, the reduction rate and competence of the CBs produced using different editing techniques on a range of standard data-sets. Knowing that NUN "Nearest Unlike Neighbor" is a concept which started from the idea that examples belonging to different classes are close only if they are close to the borders of the respective classes. For a base with Nc classes, Nc - 1 NUN (one per class) of each example are identified. Subsequently, the NUN of all initial base examples is gathered. The NUN Subsets are created which can be seen as optimal borders representatives between the classes belonging to the base. The CNN algorithm was the first reduction technique for the reference base size, based on static considerations [6]. The algorithm aims at reducing the entire input space into a representative subspace with the same properties.

Four different editing techniques are compared for this experimental study 1) CNN– the standard CNN approach; 2) NUN – CNN with cases ordered according to their NUN distances; 3) RC – CNN with cases ordered according to their relative coverage values; 4) CM with cases ordered according to their CM values and according to the following associated algorithm.

```
O-SET ← Original training example
E-SET  ← {}
Changes ← true
While Changes Do
        Changes ← false
        For each case C ∈ O-SET Do
                If E-SET cannot solve C Then
                        Changes ← true
                        Add C to E-SET
                        Remove C from O-SET
                EndIf
        End For
EndWhile
```

Algorithm 2. Condensed Nearest-Neighbour Algorithm

In order to strengthen the comparison, three different CB are used. Tow, Credit (690 cases, 15 attributes and 2 classes) and Ionosphere (351 cases, 34 attributes and 2 classes), represent classification problems from the UCI Machine Learning Repository (www.ics.uci.edu/~mlearn/MLRepository.html) [3] .The third CB relates to an industrial diagnosis dedicated to an E-maintenance high board (SORMEL) [9] (69 cases, 10 attributes and 9 classes). In the latter, the class to be found is an equipment class to be repaired, and that is formalized in example form.

7 Assessment of the proposed method

In the SORMEL CB, the space is taken from the target cases represents the total CB space. Firstly, by applying the algorithm, prior to the cases suppression, the following results were obtained:

Support Group	Support Cases	Auxiliary Cases
19	49	4

Table 1. Case-Base Statics

2 inter-auxiliaries and 2 intra-auxiliaries cases were found from auxiliary category, it was listed. After suppressing the 4 auxiliary cases and supports cases we left with one support case for every support group (i.e., 30 supports cases were suppressed). This leads to the following statistics:

- Initial size of the CB = 69
- Size of CB obtained = 35
- Reduction ratio = 49, 27 %
- Competence ratio = 100%

The obtained results are promising and by applying the proposed method on the SORMEL CB, a reduced CB obtained. Indeed, the obtained CB is roughly reduced by half, keeping the same initial competence.

8 Comparative study

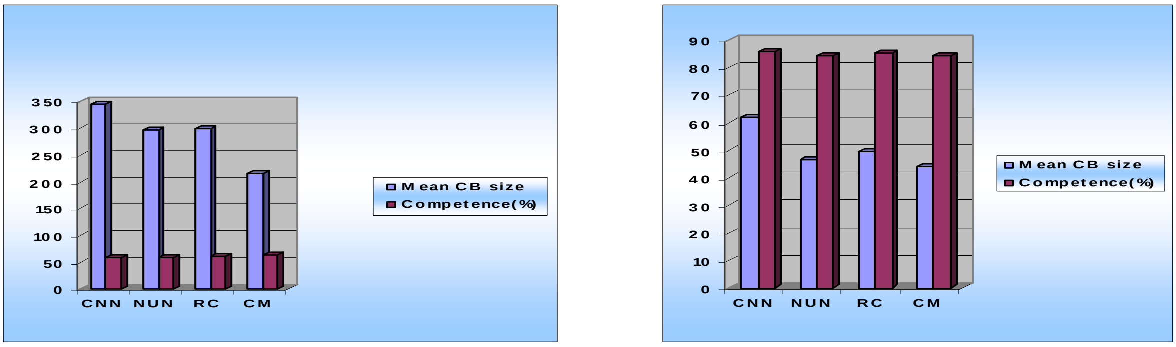
In this section, the sizes of the CBs as regards their competence on unseen target problems are compared. Each editing strategy is used to generate CBs for the 3 used data-sets. However, this time 100 random test problems are removed from the training set before CB construction. The final size of the CBs and their competence over the 100 test problems is noted. The upper value in each cell is the value of the “credit” CBs and the lower value is the value of the “Ionosphere” CBs. The results are shown in Table 2:

Property/Method	CNN	NUN	RC	CM

Reduction rate (%)	49.97 82,36	43.10 86,78	43.36 85,9	68.73 87.50
Mean CB size	344.84 61,93	297.43 46,39	299.19 49,47	215.76 43.87
Compétence (%)	58.85 85,78	58.95 84,44	60.44 85,3	62.37 84,21

Table 2. A comparison of different editing strategies over the Case-Base test in terms of mean case-base size, reduction rate and competence.

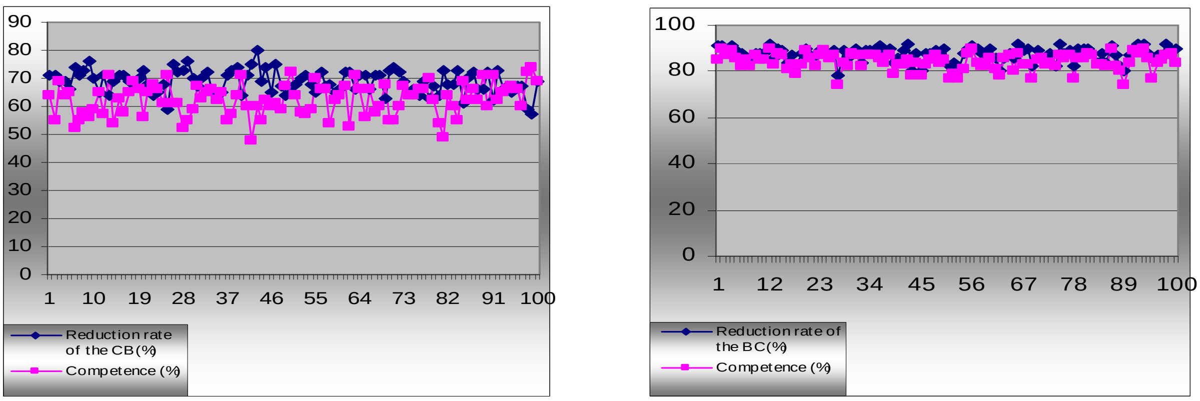
From Table 2, it can be clearly seen that the reduction rate provided by the developed approach is notably higher than the one provided by the three traditional approaches, particularly for the “Credit” data-set. The competence value is significantly better for the “Credit” data-set; however, it is almost the same as the traditional method concerning the “Ionosphere” data-set. Here, is the graph with the results from the comparison study:



(a) Credit data-set (b) Ionosphere data-set

Fig. 2. Comparison of the different strategies

As shown in figure (2), the CM method is better than the other approaches at achieving a cases reduction rate which has competence more for both data-sets “Credit” and “Ionosphere”.
Figure (3) focuses on the variation of the competence concerning the reduction ratio of the data-sets which is obtained by using the CM metric.

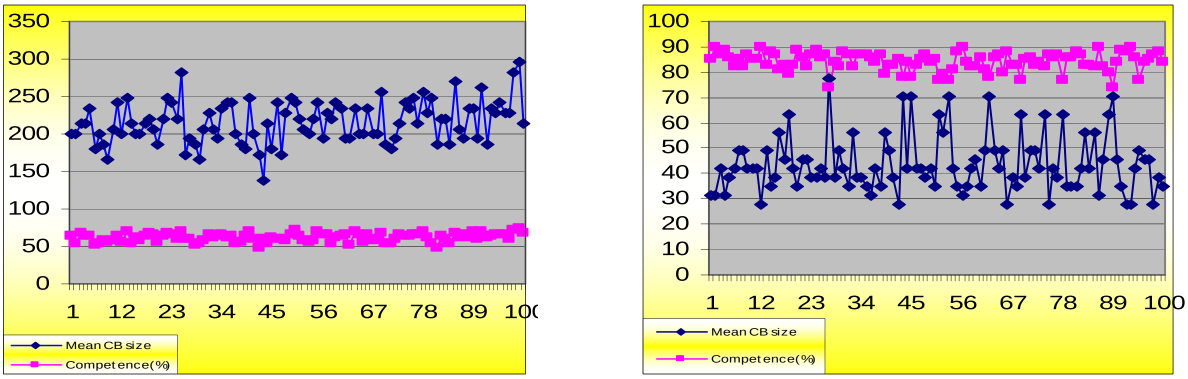


(a) Credit data-set (b) Ionosphere data-set

Fig. 3. Competence versus reduction rate with CM method

It can be seen that for both data-sets, competence can although it has keeps the same reduction rate. This implies that the competence and the size of the data-sets obtained after suppression are highly dependant on the sample chosen cases.

Figure (4) shows the variation of the competence as regards the data-sets size which is obtained by using the CM



metric.

(a) Credit data-set (b) Ionosphere data-set

Fig. 4. Competence versus size rate with CM method

It can be seen that for both data-sets, competence differs although it keeps the same reduction ratio. This implies that the competence and the size of the data-sets obtained after suppression, is highly dependant on the sample chosen cases. It is noted, that with a larger data-sets size, a larger competence value is obtained for the "Credit" base. However, the opposite is verified for the "Ionosphere" base.

9 Conclusions and future work

Three Case Based Maintenance methods (CNN, RC-CNN and NUN-CNN) were tested over three data-sets. These data-sets representing classification problems. The proposed method which used a novel algorithm combined with a Competence Metric (CM) is more efficient in terms of Case-Base reduction size and competence when applied to the studied fields.

Following this project, we plan to elaborate a method that is more complex and will merge our competence measure with a different performance measure to obtain an optimal Case Base. Finally, we will further investigate complete CBR systems.

10 References

[1] Agnar Aamodt and Enric Plaza. Case-based reasoning: Foundational issues, methodological variation, and systems. AI Communication, 7(1): 36-59, 94.
[2] Ray Bareiss, Bruce Porter, and Craig Weir. PROTOS: An Exemplar-based Learning Apprentice. Tech. Rept. AI87-53, University of Texas at Austin, 88.
[3] Catherine Blake, Eamonn Keogh, and Christopher.J. Merz. UCI Repository of machine learning databases [http://www.ics.uci.edu/~mlearn/MLRepository.html]. Irvine, CA: University of California, Department of Information and Computer Science, 98.
[4] Frans Coenen, Barry Eaglestone and Mick Ridley. Verification, Validation and Integrity in Knowledge and Database Systems: Future Directions. EUROVAV 1999: 297-312, 99.
[5] Mathieu D'aquin .Raisonnement à partir de cas décentralisé pour le Web sémantique.14th CBR workshop, Besançon, France, 06.
[6] Belur. V. Dasarathy. Nearest Neighbor Norms: NN Pattern Classification Techniques. IEEE Press, Los Alamitos, California, 91.
[7] Béatrice Fuchs and Alain Mille. Representing knowledge for case-based reasoning: the rocade system. In Enrico Blanzieri and Luigi Portinale, editors, European Conference on Case-Based Reasoning, Lecture Notes in Artificial Intelligence, number 1898, pages 86--98. Springer, Berlin, 00.
[8] Ashok Kumar Goel. Integration of case-based reasoning and model-based reasoning for adaptive design problem-solving, 89.
[9] Mohamed Karim Haouchine, Maintenance d'une base de cas dédiée au diagnostic et à sa réparation. Research master report, 05.
[10] Janet Kolodner. Case-Based Rasoning. Darpa, Clearwater, Florida, Morgan Kaufmann, San Mateo, 88.
[11] Phylis Koton. Using experience in learning and problem solving. Massachusetts Institute of Technology, Laboratory of Computer Science, PhD. Thesis MIT/LCS/TR-441, 88.

[12] Luc Lamontagne, Philippe Langlais and Guy Lapalme. Using Statistical Word Associations for the Retrieval of Strongly-Textual Cases. Special Track on Case-Based Reasoning, FLAIRS-2003, p. 124-128, Ste-Augustine, Floride, AAAI Press, 03.
[13] Luc Lamontagne and Guy Lapalme. Textual Reuse for Email Response. 7th European Conference In Case-Based Reasoning (ECCRR 2004), number. 3155, series. Lecture Notes in Computer Science, p. 242-255, Springer-Verlag, Madrid, 04.
[14] David. B. Leake and David. C. Wilson. Categorizing case-base maintenance: dimensions and directions. Advances in Case-Based Reasoning, 4th European Workshop on Case-Based Reasoning, EWCBR 98, Proceedings, Springer-Verlag, Berlin, pp. 196-207, 98.
[15] David.B. Leake and David.C. Wilson. Remembering Why To Remember: Performance-guided case-base maintenance. Advances in Case-Based Reasoning: Proceeding of EWCBR-2K, Springer-Verlag, 00a.
[16] David.B. Leake and David C. Wilson. Guiding Case-Base Maintenance: Competence and Performance. Online Proceedings of the ECAI'2000 Workshop on Flexible Strategies for Maintaining Knowledge Containers, 00b.
[17] James.T. Luxhojl and Papadias.C. TAO. Knowledge-based systems: for maintenance. Journal - American Water Works Association (J. - Am. Water Works Assoc.) ISSN 0003-150X, vol. 86, no7, pp. 54-61, 94.
[18] Maria Malek. Hybrid approaches for integrating neural networks and case based reasoning: From loosely coupled to tightly coupled models. In Soft Computing in Case Based Reasoning, editors Tharam S. Dillon Sankar K. Pal et Daniel S. Yeung, pages 73-94, 00.
[19] Shaul Markovitch and Paul David Scott. The Role of Forgetting in Learning. In Proceedings of the Fifth International Conference on Machine Learning, pages 459-465, 88.
[20] Alain Mille. Raisonnement basé sur l'expérience pour coopérer à la prise de décision, un nouveau paradigme en supervision industrielle. Ph.D. Thesis, Saint-Etienne University, 95.
[21] Alain Mille. Modèle conceptuel du raisonnement basé sur les cas. In Isabelle Bichindaritz, editor, 4th Séminaire Français sur le Raisonnement à Partir de Cas, Paris, pages 40--52, Université René Descartes (Paris V), 95.
[22] Steven Minton. Qualitative Results Concerning the Utility of Explanation-Based Learning. Artificial Intelligence, 42: 363-391, 90.
[23] Kirsti Racine and Qiang Yang. On the consistency Management of Large Case Bases: the Case for Validation. To appear in AAAI Technical Report, Verification and Validation Workshop, 96.
[24] Thomas Reinartz, Ioannis Iglezakis and Thomas Roth-Berghofer. Review and restore for Case Base Maintenance. Computational Intelligence Journal, 17(2), 01.
[25] Tae Hyup Roh, Kyong Joo Oh, and Ingoo Han. The collaborative filtering recommendation based on some clustering-indexing cbr. In Expert Systems with Applications, Elsevier Science, volume 25, pages 413-423, 03.
[26] Thomas Roth-Berghofer and Ioannis Iglezzakis. Six Steps in Case–Based Reasoning: Towards a maintenance methodology for case–based reasoning systems. Includes Proceedings of the 9th German Workshop on CBR, GWCBR, Germany, 01.
[27] Barry Smyth and M.T. Keane. Remembering To Forget: A competence Preserving Deletion Policy for Case-Based Reasoning Systems. In: Proceeding of the 14th International Joint Conference on Artificial Intelligent, Morgan-Kaufmann. 377-382, 95.
[28] Barry Smyth and Pádraig Cunningham. The Utility Problem Analysed: A Case-Based Reasoning Perspective. Third European Workshop on Case-Based Reasoning, Lausanne, Switzerland, 96.
[29] Barry Smyth, Case-base maintenance. Tasks and Methods in Applied Artificial Intelligence. 11th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems, IEA-98-AIE. Proceedings. Springer-Verlag, Berlin, Germany, pages 507-516 vol.2, 98a.
[30] Barry Smyth and Elizabeth McKenna. Modelling the Competence of Case-Bases. Advances in case-based reasoning, Lecture notes in computer science, Dublin, 1488: 208-220, 98b.
[31] Barry Smyth and Elizabeth McKenna. Building Compact Competent Case-Bases, Case-based reasoning research and development. See on Monastery, Lecture notes in computer science, 1650: 329-342, 27-30, 99a.
[32] Barry Smyth and Elizabeth McKenna. Footprint-Based Retrieval. In Proceedings of the Third International Conference on Case-Based Reasoning, ICCBR '99, Springer Verlag, Berlin, Germany, 27-30, 99b.
[33] Barry Smyth and Elizabeth McKenna. Competence models and the maintenance problem. Computational Intelligence: Special Issue on Maintaining Case-Based Reasoning Systems, In Press, 02.
[34] Qiang Yang and Jing Wu. Keep it simple: A case-base maintenance policy based on clustering and information theory. 13th Biennial Conference of the Canadian Society for Computational Studies of Intelligence, AI 2000: advances in artificial intelligence, Proceedings Lecture Notes in Artificial Intelligence, Montréal PQ, Springer-Verlag, Berlin, Vol.1822. 2000: 102-114, 14-17, 00.
[35] Qiang Yang and Jun Zhu. A case addition policy for case-base maintenance. Computational Intelligence Journal, A Special Issue on Case-Base Maintenance, Boston MA UK, Blackwell Publishers, 17(2): 250-262, 01.